# 利用帕累托非支配关系实现高效三目标差分进化的方法 *

许玉龙 [1,2,3]，潘　旭 [1]，王忠义 [1†]，盛梦园 [1]，王林景 [1]

(1. 河南中医药大学 信息技术学院，郑州 450046; 2. 郑州大学 信息工程学院，郑州 450052; 3. 香港科技大学 计算机科学与工程系，香港)

**摘　要：**在多目标进化算法中，时间复杂度过高是普遍的问题，特别是三个目标函数以上时，解的等级分配占用了过多运算时间。针对三目标问题，利用帕累托支配关系，对解的等级分配进行研究，发现经典的等级排序及分配方法存在一定冗余操作，需对全部的解先排序后，才能再分配等级并选择下一代，造成部分不必要的运算。为减少该冗余，利用帕累托非支配关系结合差分进化，实现高效三目标进化算法。算法每次迭代对种群中最高等级的个体进行计算，在分配等级同时进行选择后代个体操作，当后代种群生成时便跳出计算，从而减少个体的计算数量，降低运算量，同时给出该方法的相关理论分析和证明过程。然后，针对一系列三目标优化问题，将提出方法与著名排序方法 NSGAII，及近年来优秀的 ENS 方法进行对比实验。仿真实验结果表明，提出方法在时间复杂度和收敛速度上优于经典方法，稍差于 ENS 方法。在标准测试函数 DTLZ1-DTLZ6 的性能上，提出方法近似于 ENS 方法，优于 NSGAII 算法，从而验证了提出方法的有效性和正确性。

**关键词：**三目标进化；帕累托；非支配解排序；收敛速度

**中图分类号：**TP391　　doi: 10.3969/j.issn.1001-3695.2017.09.0909

# Using Pareto dominance relationship to realize efficient tripe-objective differential evolution algorithm

Xu Yulong[1, 2, 3], Pan Xu[1], Wang Zhongyi[1†], Sheng Mengyuan[1], Wang Lingjing[1]

(1. *Institute of Information & Technology Henan University of Chinese Medicine*, *Zhengzhou* 450046, *China*; 2. *Institute of Information Engineering*, *Zhengzhou University*, *Zhengzhou* 450052, *China*; 3. *Department of Computer Science & Engineering Hong Kong University of Science & Technology Hong Kong*, *China*)

**Abstract:** In the Multi-objective evolutionary algorithm, the solution of the rank distribution takes up most of the computation time. According to the Pareto dominance relationship, we research the solution of the rank and distribution in this paper, and we found that there are some redundancy operation of rank and allocation in the classic methods. The classic algorithms need to rank all solutions, and select the next generation after sort, so it caused some unnecessary sort operations. To reduce the redundancy, we introduce a fast method of non-dominated sorting. The proposed algorithm only deals the individual which belong the highest level in current population, and meanwhile it can choose individual into the next generation during the process of distribution level. When the next generation individuals were selected enough, the program is ended. This method has reduced the number of the sorting of individuals, and the time complexity is also reduced greatly. Then, we use three objectives optimization problem as an example, combine the fast non-dominated sorting method with the differential evolution algorithm, and propose a fast three objective differential evolution algorithm based on non-dominated sorting. Finally, we compare our method with the famous ranking method and ENS/BNS method. The simulation results show that our proposed algorithm in time complexity and convergence speed is better than the classical method, and is not worse than the ENS/BNS algorithm. On the performance of the standard test function DTLZ1-DTLZ6, the proposed method is similar to or better than the comparison algorithm. Based on above experiment, we have verified the correctness and effectiveness of our proposed method.

**Key Words:** tripe-objective; Pareto; non-dominated solution sorted; convergence speed



## 0　引言

多目标优化问题 (multi-objective optimization problems, MOP)的目标函数一般多于两个，其解是一个集合，称为非支配解集(nondominated set)或 Pareto 最优解集 (Pareto-optimal set) [1]。有多种方法可以求解一般优化问题，如线性规划、梯度下降、进化算法等，但相关成果表明，多目标进化算法(multi-objective evolutionary algorithms, MOEAs)较适宜求解多目标优化问题[1,2]。

多目标进化受到国际学者的广泛关注，大量优秀的算法被

**基金项目：**国家自然科学基金资助项目（81703946，61772475）；河南省科技攻关研究项目（172102210361，172102310536）；河南省高校重点科研项目（15A520083，16A520060，17B520017）；河南中医药大学博士基金资助项目（BSJJ2015-19）

**作者简介：**许玉龙（1984-），男，河南许昌人，讲师，博士，主要研究方向为生物进化算法及其应用；潘旭（1996-），男，河南驻马店人，本科生，主要研究方向为进化计算及其应用；王忠义（1983-），男（通信作者），河南兰考人，讲师，硕士，主要研究方向为智能信息处理（flyxyl@126.com）；盛梦园（1995-），女，河南商丘人，本科生，主要研究方向为智能信息处理；王林景（1985-），女，河南南阳人，硕士，主要研究方向为智能优化算法.

提出，其中以 NSGAII 算法[2]和 SPEAII 算法[3]最为著名。上述两种算法都是通过空间密度和帕累托支配来判断解的好坏，通过构造帕累托候选解集来获得最优解。国内学者中，焦李成、陈火旺、崔逊学、郑建国、蔡自兴等人[4~8]，对多目标进化的主要技术、理论分析及发展历史进行深入研究，并指明了研究方向和面临的问题。其中，研究提高算法计算效率是一个非常有意义的问题。

为提高算法的计算效率，学者们进行了不同的尝试，NSGAII 算法[2]相比上一代版本及同时期其他著名算法，在计算速度上有较大优势，但其全局时间复杂度仍为 $O(n^2)$。

Tang 等人[9]依据帕累托支配关系对该问题进行改进，使用擂台原则对得到的解分配等级，从种群中随机选择一个解，把它视为“擂主”，然后用剩余的解与它进行支配关系比较，若种群中某个解支配“擂主”，用解替代原来的“擂主”，算法的时间复杂度和空间复杂度分别为 $O(mn^2)$和 $O(n)$。在仿真实验中，该方法展示出比 NSGAII 更高的计算效率。同样为提高计算效率，Clymont 等人[10]提出了著名的非支配排序算法：deductive sort，该方法能记录所有解之间的比较结果，推断所有解的支配关系，从而避免多余的比较，算法的时间复杂度和空间复杂度分别小于 $O(mn^2)$和 $O(n)$，实验结果验证了该方法的有效性。

不久前，张兴义等人[11]提出一种高效的非支配解排序方法，命名为 ENS。该方法最大特点是：当一个候选解与已分配等级的解比较时，得出此候选解的等级，避免了大量冗余操作。算法首先对第一个目标函数值按升序排序，此后，候选解 $p_n$ 与 $F$($F$ 为当前检查等级)中的解只有两种支配关系：候选解被等级 $F$ 中的解支配；候选解不被等级 $F$ 中的解支配。作者基于 ENS 提出了 ENS-SS 和 ENS-BS 两种排序算法。在 ENS-SS 中，从第一个等级开始检查，直到 $p_n$ 被分配等级结束，如果 $p_n$ 被 $F$ 中任意解支配，则继续检查后续等级；如果 $p_n$ 不被 $F$ 中所有解支配，将 $p_n$ 分配到 $F$ 中；而在 ENS-BS 中，从中间等级开始，检查范围初始化为已存在等级。每检查完一个等级，范围缩减一半。如果 $p_n$ 被 $F$ 中任意解支配，检查范围缩减为 $F$ 后的所有等级；如果 $p_n$ 不被 $F$ 中所有解支配，检查范围缩减为 $F$ 之前所有等级，直到 $p_n$ 被分配到某个等级中。实验表明，ENS 算法在计算效率上显著优于对比算法。

差分进化算法(differential evolution, DE)[12,13]是近年来涌现出的优秀算法之一，它以高效、简便的特性受到了众多学者关注。文献[14]较早介绍了使用 DE 结合非支配关系求解多目标优化问题，并取得较好的效果。Robic 等人提出了基于差分进化最著名的多目标算法 DEMO[15]。杨平等人[16]使用选举法则来构造帕累托非支配解集，并进行理论证明其能够提高效率，取得了理想的成果，但其成果仅是对非支配排序次数的比较。任长安等人[17]提出一种改进的基于目标空间分割的多目标进化算法，算法基于区间索引值进行高效的运算，但该方法仅在两目标问题上进行了测试。类似的，学者们对差分进化算法及其在实际中的应用也做了较多研究[18~22]。不久前，文献[23]结合聚类和数据分析的方法对多目标差分进化算法进行改进、文献[24,25]提出基于网格索引指导进化方向的多目标差分进化算法，都取得了较好的效果。但是，上述多目标差分进化方法在对非支配排序时，需要对全部的个体分配等级，会造成部分冗余操作。为改进该问题，本文前期对两目标差分进化算法做了相关的研究[26]，介绍了一种快速的非支配解排序方法，并得到了较好的效果，但该方法只是对两目标优化问题的研究。

因此，本文在已有工作的基础上，介绍利用帕累托非支配关系实现高效三目标差分进化的方法，针对三目标优化问题，该方法能够在进行非支配解排序同时，选择个体进入下一代。在实验对比测试中，该方法与经典和最新算法进行对比研究，理论分析和实验结果都证明了提出方法的正确性和高效性。

## 1　相关理论

### 1.1　差分进化算法

差分进化算法自提出以来,凭借速度快、易实现、参数少等特点，迅速成为进化计算领域中最著名的算法之一。差分进化属于启发式算法，包括了变异、交叉、选择、更新等基本操作，为节省篇幅，下面简要介绍其基本流程。

首先随机产生初始种群，然后依该种群进行变异操作，有多种变异策略可供使用，常见的有：

a) 随机引导变异 DE/rand/1：

$$v_{i,g} = x_{r1,g} + F_i \cdot \left( x_{r2,g} - x_{r3,g} \right) \tag{1}$$

b) 最优引导变异 DE/ best/1：

$$v_{i,g} = x_{best,g} + F_i \cdot \left( x_{r1,g} - x_{r2,g} \right) \tag{2}$$

c) 当前到随机引导变异 DE/current-to-best/1：

$$v_{i,g} = x_{i,g} + F_i \cdot \left( x_{best,g} - x_{i,g} \right) + F_i \cdot \left( x_{r1,g} - x_{r2,g} \right) \tag{3}$$

其中:$x$ 是目标向量，$v$ 是变异向量，下标 $i$ 表示第 $i$ 个个体，$i$=1,2,3,…,$NP$,$NP$ 是种群数量，$g$ 表示第 $g$ 代种群。式(3)中 $x_{best,g}$ 是种群中最优个体，$x_{i,g}$ 是当代个体，$v_{i,g}$ 是变异后个体，$x_{r1,g}$, $x_{r2,g}$, $x_{r3,g}$ 为群体中随机选择三个互不相同的个体，($x_{r1,g}$- $x_{r2,g}$)即为差分向量，$F$ 是缩放因子，用于对差分向量缩放，控制搜索步长。

然后，进行交叉操作，把生成的变异向量 $v_{i,g}$ 与个体向量 $x_{i,g}$ 进行交叉，得到实验向量 $u_{i,g}$。差分进化的交叉方式包括二项式交叉和指数交叉，一般情况，常用的是二项式交叉，方法如下：

$$u_{i,G}^{j} = \begin{cases} v_{i,g}^{j} & \text{if } rand_{i,j}[0,1] \le Cr \text{ or } j = j_{rand} \\ x_{i,g}^{j} & \text{otherwise} \end{cases} \tag{4}$$

其中：$u_{i,G}^{j}$ 表示 $G$ 代种群中第 $i$ 个体的第 $j$ 个基因,$j$=1,…,$D$,$D$ 是问题的维数。$Cr \in [0,1]$，是交叉概率因子，控制算法收敛的速度。随机生成数 $rand_{i,j}$ 小于 $Cr$ 或者 $j$ 等于 $j_{rand}$ 时，实验向量从变异向量继承，否则，从目标向量继承。$j_{rand}$ 是[1,$D$]内的随

机数，它保证实验向量 $U$ 中至少有一个基因来自变异向量。

接下来是选择操作，差分进化采用高强度贪婪的选择策略，方法是将目标向量与实验向量进行逐一比较，优秀者进化到下一代。以求解最小化为例，其选择方法如式(5)所示。

$$x_{i,g+1}=\begin{cases}u_{i,g} & \text{if } f\left(u_{i,g}\right)\le f\left(x_{i,g}\right)\\ x_{i,g} & \text{otherwise}\end{cases} \tag{5}$$

以上概括了差分进化算法的简要流程，其详细介绍参见文献[12]。

### 1.2　多目标优化概述

在多目标优化中，各个目标函数之间是相互冲突、不可调和的，问题的最优解通常是一个解集而不是一个解。而在单目标优化中，最优解是唯一的，所以多目标优化与单目标优化有本质的区别。

多目标优化过程中，两个解之间的占优关系又称为帕累托 Pareto 支配关系，多目标优化问题不存在全局最优解，会存在多个 Pareto 最优解，这些 Pareto 最优解构成的集合，就称为问题的最优解集。解集中的所有解之间都是非支配关系，决策者依据自己的爱好和需求，从中挑选一个或多个作为最终的最优解使用。多目标进化算法就是要得到尽量多的、分布均匀的、接近理论解集的 Pareto 最优解，关于多目标进化相关的概念和数学描述参见文献[5]，为节省空间在此不再赘述。

## 2　基于帕累托非支配关系实现高效三目标差分进化

### 2.1　高效的非支配解排序方法

NSGAII 是最著名的多目标优化算法之一[2]，许多改进算法都参考了它的思想和过程。而本文发现该算法存在计算冗余，算法中的非支配解排序过程需对所有个体分配等级后，排序操作才结束，然后才能进行选择下一代操作。最近被提出的著名算法 ENS[11]中，也需要对所有个体先分配等级，但 ENS 避免了所有非必需和部分必需的支配比较次数，提高了算法运行速度。上述算法中，都是先对种群所有染色体进行等级确定，然后按个体等级由高到低排序，从等级 1 到等级 $n$ 的顺序选出染色体进入下一代。

在多目标进化中，假设种群规模为 N，需要选择进入下一代的染色体数量是即是 N，而经典算法都是对 2N 个染色体计算，需要对所有 2N 个染色体进行排序、等级分配、判断比较，然后选择出 N 个进入子代。设想在排序和等级分配时，能否仅对部分 2N 个体进行计算，就能找到满足进入下一代的 N 个染色体。这将使参与计算的个体数量下降，算法的运行效率便有所提高。因此，依据三目标优化中非支配解排序的潜在规律，本文提出高效的三目标非支配解排序方法，该方法不需要对所有 2N 个体进行计算对比，能够降低时间复杂度，具体算法参见算法 1.

**算法 1**　高效的三目标非支配解排序方法

**输入：**合并后的初始种群和变异种群 U，染色体个数为 2 倍的种群

**输出：**满足进入下一代种群的染色体，个数为 N

for 等级 rank=1
　for 个体 i=1:2N
　　找到属于第 *rank* 级的 4 个特殊个体,存储到集合 *Srank*
　　这 4 个分别是到 $f_1$ 轴、 $f_2$ 轴、$f_3$ 轴、原点距离最小的个体
　　个别情况下，4 个个体之间可能会重复
　　对集合 *Srank* 中个体分配等级 *rank*
　　集合 $U$ 与集合 *Srank* 做差集,得到 $U'$
　　for 个体 j=1: U’
　　　*if* 个体 *j* 与集合 *Srank* 中任一个体都非支配
　　　　则 *j* 等级为 *rank*,将 *j* 存储到 *Srank* 集合
　　　*end if*
　　end for 个体 j=1: U’
　　得到第 *rank* 等级全部个体，存于集合 *Srank*
　　将集合 *Srank* 中个体全部进入子代
　　*if* 进入子代个体总数 *sum*>*N*
　　　溢出情况发生，进行拥挤距离计算，选出部分个体
　　end if
　　*if* 进入子代个体总数 *sum*==N
　　　结束程序,得到子代个体
　　end if
　*end for* i=1:2N
　集合 $U$ 与 *Srank* 做差集,结果记为 $Q$ 集合
　令 $U=Q$,即用 $Q$ 更新 $U$
　*rank*++,考虑计算下一个等级个体
end rank

算法 1 中，令种群规模 $N$，算法输入为 2 倍的种群规模集合 $U$，输出为选出 $N$ 个染色体进入子代。提出的排序方法在选择下一代个体时，无须对所有的 2N 个体进行计算对比，减少了个体比较次数，避免了存在的计算冗余。主要特点如下：首先计算得到当前种群中，属于最高等级的 4 个特殊个体（假设不重合），将他们存储到集合 *Srank*。然后，将种群剩余个体与 *Srank* 中个体逐一比较，当个体 $i$ 被集合 *Srank* 中的某个体支配时，结束与集合中剩余个体的比较，节省了比较次数。当个体 $i$ 与集合 *Srank* 中所有个体都是非支配关系时，将个体 $i$ 加入集合 *Srank*。在比较时借鉴文献[11]的经验：从集合 *Srank* 中最后个体开始，使得每次更新 *Srank* 后，其个体都是按第一个目标函数值升序排列，所以 *Srank* 中靠后个体更可能支配个体 $i$。重复该操作进行处理，最终得到当前种群中最高等级的所有个体，等级记为 *rank*=1，存储在集合 *Srank*。集合 *Srank* 中个体有待进入下一代，如果集合中个体全部进入下一代后，下一代个体总数超过 $N$，则进行拥挤距离计算选择部分个体；如果下一代

个体总数不超过 $N$，则所有个体进入下一代，并将 *rank* 增 1，更新种群 $U$，重复上述过程，选择出第二等级的个体；如果下一代个体总数等于 $N$，则程序结束，不需要对剩余的所有个体进行计算，提高了计算效率。根据一般经验，到进化过程的中期，算法只需计算前几个等级个体，便可满足下一代的选取，从而不需再对剩余个体进行计算，较大地节省了时间。

## 2.2　高效三目标非支配解排序算法理论基础

提出的高效三目标非支配解排序算法中，最重要的是确定四个特殊的个体，并保证这些个体属于当前种群中最高等级。需要说明的是，对于三目标优化而言，最多只有 4 个特殊个体，由于这些特殊个体可能会重合，所以特殊个体数目具体应为 1 至 4 个，在此本文假设一般情况他们不重合，所以需确定为 4 个个体。如何证明这些个体在当前种群中属于最高等级，是较为关键的基础，证明过程如下：

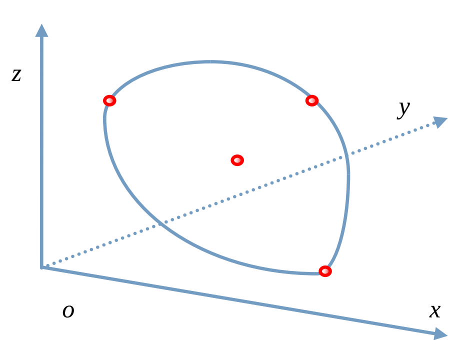



图 1　高效三目标非支配解排序中四个特殊个体

根据相关理论，给出如下推理。为描述方便，如图 1 所示，本文使用 $x$、$y$、$z$ 分别代替 $f_1$、$f_2$、$f_3$。

推理 1　若存在个体 $a(x,y,z)$的 $x$ 坐标值，是当前种群中所有 $x$ 坐标值的最小值，则 $a$ 是种群最高等级中的个体。

推理 2　若存在个体 $b(x,y,z)$的 $y$ 坐标值，是当前种群中所有 $y$ 坐标值的最小值，则 $b$ 是种群最高等级中的个体。

推理 3　若存在个体 $c(x,y,z)$的 $z$ 坐标值，是当前种群中所有 $z$ 坐标值的最小值，则 $c$ 是种群最高等级中的个体。

推理 4　若存在个体 $d(x,y,z)$到原点距离的值，是当前种群中任意个体到原点距离值的最小值，则 $d$ 是种群最高等级中的个体。

证明

采用反证法，已知当前种群，所有个体中 $x$ 的坐标值最小为 $x_{min}$，如果个体 $a$ 坐标是 $a(x_{min},a_y,a_z)$,则当前种群中不存在一个个体 $l$，使得 $l$ 的横坐标 $l_x$ 小于 $x_{min}$。

本文假设，存在一个个体 $l$，使得该个体支配个体 $a(x_{min},a_y,a_z)$，则根据 Pareto 相关原理，可得到 $l_x<x_{min}$, $l_y<a_y$ 且 $l_z<a_z$,明显，该结论与已知条件矛盾，即不存在个体 $l$ 支配 $a$，则得出个体 $a$ 属于当前最高等级个体，证毕。其他推理 2、3 和 4 的证明过程与此类似。由此可知，提出方法中首先确定 4 个特殊个体，一定是当前种群中最高等级的个体。

## 2.3　高效三目标差分进化算法框架和时间复杂度分析

依据提出的高效三目标非支配解排序方法，结合差分进化，提出一种基于非支配解排序的高效三目标差分进化算法(FMODE)。FMODE 的总体流程如算法 2 所示。

**算法 2**　高效三目标差分进化流程

**输入**：算法参数、种群规模 N、进化代数 FES

**输出**：进化结束后的解集

种群初始化，$X(0)$ :$x_1(0),\ldots, x_n(0)$;
加载算法参数 $G, N,Cr ,F$.
*while*(进化开始不满足结束条件)
　变异，依据式(1)得到 $M$
　交叉，依据式(4)得到 $Q$
　合并 $M$ 、$Q$，得到 $U$，$U$ 含有 $2N$ 个个体
　从 $U$ 中选择满足条件的 $N$ 个个体进入下一代
　使用提出的高效三目标算法，参见算法 1
　　算法 1 找到第 *rank* 等级的个体时,令为 $K$
　　将 $K$ 个个体设为选择进入下一代个体时
　　*if* 已选择个体总数等于 $N$
　　　*break*，进化代数加 1
　　*if* 已选择个体总数小于 $N$
　　　$K$ 个个体进入下一代
　　*if* 已选择个体总数大于 $N$
　　　对第 *rank* 个等级进行拥挤距离计算
　　　选择合适的个体，进入下一代，使得下一个个体为 $N$ 个
　　　*break*, 进化代数加 1
end while

提出算法在非支配排序时，仅处理当前种群中最高等级个体，同时检测是否选足下一代个体，当发现选够时，即终止算法，大幅减少了运算时间。而经典非支配解排序算法需要对所有个体进行排序，当所有个体都分配等级后，再选择下一代，所以用时较多。第一代 NSGA 算法的时间复杂度是 $O(mn^3)$，虽然 Deb 之后提出了改进的 NSGAII，但排序时间复杂度是 $O(mn^2)$，原因是算法在运行时，要对所有个体分配等级后，才能再选出下一代个体，存在冗余操作。

著名算法 ENS[11]在排序时，有两个主要步骤：1)把种群按第一个目标函数值升序排列，时间复杂度为 $O(nlog^n)$；2)对每个解分配等级。ENS-SS 在非支配解排序时，最坏情况是种群中所有个体都属于同一等级，此时时间复杂度是 $O(mn^2)$，最好情况是种群个体属于 $\sqrt{n}$ 个等级左右，时间复杂度是 $O(mn\sqrt{n})$。在 ENS-BS 进行非支配解排序时，最坏情况和 ENS-SS 相同，时间复杂度是 $O(mn^2)$，最好的情况是所有解的等级都不相同，时间复杂度为 $O(mnlog^n)$。

本文提出的算法中，最好情况是第一等级个体数量 $\geq N$。等于 N 时,直接将该等级所有个体进入下一代；大于时，进行拥挤距离操作，选择部分个体进入下一代，剩余所有个体不再进行

等级分配,所需时间复杂度是 $O(mn)$，算法按目标函数值排序的时间复杂度为 $O(nlog^n)$，计算到原点距离时间复杂度为 $O(n)$，故总时间复杂度为 $O(mn)+2*O(nlog^n)+O(n)$。最坏情况下，本文算法与 NSGAII 相同，要对所有个体分配等级，时间复杂度为 $O(mn^2)$，总时间复杂度为 $O(mn^2)+2*O(nlog^n)+O(n)$。在进化初始阶段，初始种群个体分布较为随机，将要处理多个等级才能选足个体进入下一代。但是，在进化中后期，从前几个等级就能选足下一代，从而提高了计算效率。

## 3　仿真实验

### 3.1　实验环境及参数设置

为了测试算法 FMODE 的有效性，采用国际标准测试问题 DTLZl-DTLZ6，这些问题的 Pareto 前沿具有一定代表性，具体函数描述如表 1 所示。FMODE 在实验中参数的取值如下：种群规模 $N$=200,变异概率 $F$=0.5，交叉概率 $Cr$=0.2，由于不同函数中接近理论最优解时所需的进化代数不同，设置不同的进化代数如表 1 所示，选变异方式为“DE/Rand/1/bin”。对比算法为著名的 NSGAII 和 ENS 算法，他们的参数依据算法的原文献进行设置。实验环境：操作系统 Win10，处理器 Intel(R) CoreTM i5-4590CPU@ 3.3GHz，软件平台 Matlab R2015b。由于 IGD(inverted generational distance)[23]包含了多样性和收敛性，且能体现算法性能，所以本文选用 IGD 作为测试指标，详见式（11）所示。$P^*$代表 Pareto 理论最优解集，$P$ 代表求得的最优解集，$v \in P$，$d(v,P)$代表 $v$ 和 $P$ 之间的最小欧拉距离。

$$IGD\left(P^*,P\right)=\frac{\sum_{v\in P^*} d\left(v,P\right)}{\left|P^*\right|} \tag{11}$$

表 1　6 个三目标测试函数

| 问题 | 取值范围 | 目标函数 | 维度 |
|---|---|---|---|
| DTLZ1 | $x\in[0,1]$ | $f_1(x)=\frac{1}{2}x_1x_2\ldots x_{m-1}(1+g(x_m))$<br>$f_2(x)=\frac{1}{2}x_1x_2\ldots(1-x_{m-1})(1+g(x_m))$<br>$\vdots$<br>$f_{m-1}(x)=\frac{1}{2}x_1(1-x_2)(1+g(x_m))$<br>$f_m(x)=\frac{1}{2}(1-x_1)(1+g(x_m))$<br>$g(x_m)=100\left[\lvert x_m\rvert+\sum_{x_i\in x_m}\left((x_i-0.5)^2-\cos(20\pi(x_i-0.5))\right)\right]$ | m+4 |
| DTLZ2 | $x\in[0,1]$ | $f_1(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}\pi}{2}\right)\cos\left(\frac{x_{m-1}\pi}{2}\right)$<br>$f_2(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}\pi}{2}\right)\sin\left(\frac{x_{m-1}\pi}{2}\right)$<br>$f_3(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\sin\left(\frac{x_{m-2}\pi}{2}\right)$<br>$\vdots$<br>$f_{m-1}(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\sin\left(\frac{x_2\pi}{2}\right)$<br>$f_m(x)=(1+g(x_m))\sin\left(\frac{x_1\pi}{2}\right)$<br>$g(x_m)=\sum_{x_i\in x_m}(x_i-0.5)^2$ | m+9 |
| DTLZ3 | $x\in[0,1]$ | $f_1(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}\pi}{2}\right)\cos\left(\frac{x_{m-1}\pi}{2}\right)$<br>$f_2(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}\pi}{2}\right)\sin\left(\frac{x_{m-1}\pi}{2}\right)$<br>$f_3(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\sin\left(\frac{x_{m-2}\pi}{2}\right)$<br>$\vdots$<br>$f_{m-1}(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\sin\left(\frac{x_2\pi}{2}\right)$<br>$f_m(x)=(1+g(x_m))\sin\left(\frac{x_1\pi}{2}\right)$<br>$g(x_m)=100\left[\lvert x_m\rvert+\sum_{x_i\in x_m}\left((x_i-0.5)^2-\cos(20\pi(x_i-0.5))\right)\right]$ | m+9 |

续表

| 问题 | 取值范围 | 目标函数 | 维度 |
| --- | --- | --- | --- |
| DTLZ4 | $x\in[0,1]$ | $f_1(x)=(1+g(x_m))\cos\left(\frac{x_1^\alpha\pi}{2}\right)\cos\left(\frac{x_2^\alpha\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}^\alpha\pi}{2}\right)\cos\left(\frac{x_{m-1}^\alpha\pi}{2}\right)$<br>$f_2(x)=(1+g(x_m))\cos\left(\frac{x_1^\alpha\pi}{2}\right)\cos\left(\frac{x_2^\alpha\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}^\alpha\pi}{2}\right)\sin\left(\frac{x_{m-1}^\alpha\pi}{2}\right)$<br>$f_3(x)=(1+g(x_m))\cos\left(\frac{x_1^\alpha\pi}{2}\right)\cos\left(\frac{x_2^\alpha\pi}{2}\right)\ldots\sin\left(\frac{x_{m-2}^\alpha\pi}{2}\right)$<br>$\vdots$<br>$f_{m-1}(x)=(1+g(x_m))\cos\left(\frac{x_1^\alpha\pi}{2}\right)\sin\left(\frac{x_2^\alpha\pi}{2}\right)$<br>$f_m(x)=(1+g(x_m))\sin\left(\frac{x_1^\alpha\pi}{2}\right)$<br>$g(x_m)=\sum_{x_i\in x_m}(x_i-0.5)^2,\ \alpha=100$ | $m+9$ |
| DTLZ5 | $x\in[0,1]$ | $f_1(x)=(1+g(x_m))\cos\left(\frac{x_1\pi}{2}\right)\cos\left(\frac{x_2\pi}{2}\right)\ldots\cos\left(\frac{x_{m-2}\pi}{2}\right)\cos\left(\frac{x_{m-1}\pi}{2}\right)$<br>$f_2(x)=(1+g(x_m))\cos\left(\frac{\theta_1\pi}{2}\right)\cos\left(\frac{\theta_2\pi}{2}\right)\ldots\cos\left(\frac{\theta_{m-2}\pi}{2}\right)\sin\left(\frac{\theta_{m-1}\pi}{2}\right)$<br>$f_3(x)=(1+g(x_m))\cos\left(\frac{\theta_1\pi}{2}\right)\cos\left(\frac{\theta_2\pi}{2}\right)\ldots\sin\left(\frac{\theta_{m-2}\pi}{2}\right)$<br>$\vdots$<br>$f_{m-1}(x)=(1+g(x_m))\cos\left(\frac{\theta_1\pi}{2}\right)\sin\left(\frac{\theta_2\pi}{2}\right)$<br>$f_m(x)=(1+g(x_m))\sin\left(\frac{\theta_1\pi}{2}\right)$<br>$\theta_i=\frac{\pi}{4(1+g(x_m))}(1+2g(x_m)x_i),\forall i=2,3,\ldots,(m+1)$<br>$g(x_m)=\sum_{x_i\in x_m}(x_i-0.5)^2$ | $m+9$ |
| DTLZ6 | $x\in[0,1]$ | $f_1(x)=x_1$<br>$f_2(x)=x_2$<br>$\vdots$<br>$f_{m-1}(x)=x_{m-1}$<br>$f_m(x)=(1+g(x_m))h(f_1,f_2,\ldots,f_{m-1},g)$<br>$g(x_m)=1+\frac{9}{\lvert x_m\rvert}\sum_{x_i\in x_m}x_i$<br>$h(f_1,f_2,\ldots,f_{m-1},g)=m-\sum_{i=1}^{m}\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right]$ | $m+9$ |

### 3.2　运行速度对比

为验证提出排序方法的高效性，本节给出与 NSGAII、ENS 非支配解排序方法在运行速度和比较次数上对比，结果如表 2、3 所示。表中时间单位是 s，比较次数单位是次，在不同种群规模下分别测试，测试问题都是三目标优化问题。算法 ENS_SS 和 ENS_BS 分别代表的是采用 DE 进化框架结合 ENS_SS 排序和 ENS_BS 排序。需要说明的是，在本节测试中，四种算法对种群中所有个体仅计算等级操作，不做拥挤距离计算。即使对所有个体都计算等级处理，本文算法每次得到当前最高等级个体，在后续等级分配中，则不需要再和已分配过等级的个体对比，理论上与 ENS 相差不大，但比 NSGAII 效率更高。

在表 2 和 3 中，NP 为种群个数，F 为特定等级个数，括弧外是排序耗时，单位为 s，括弧内是比较次数，表 2 使用具有特定等级个数的种群，并设置等级个数为 $\sqrt{NP}$ 左右，在此等级下 ENS-SS 表现出最优性能,详见文献[11]。

通过表 2 和 3 可以看出，本文的排序算法在个体数量较少时排序耗时与其他三种算法相差不大。当种群个体较多时，本文算法排序耗时略多于 ENS 方法，但远小于 NSGAII 的排序耗时。简言之，FMODE 表现计算效率略逊于 ENS 算法，但都显著优于 NSGAII 算法。由于 ENS 是近年来提出的著名算法，与其对比实验结果显示出其的优越性，也说明了提出算法在计算速度上的高效性。

表 2 四种排序方法对于具有 $\sqrt{NP}$ 等级种群的对比测试:排序耗时秒(比较次数)

| 算法 | NP, F | | | | | |
|---|---|---|---|---|---|---|
| | 100, 10 | 500, 22 | 1000, 32 | 2000, 45 | 5000, 71 | 10000, 100 |
| NSGAII | 0.064292 | 1.2045 | 4.7786 | 19.7604 | 143.8898 | 713.4332 |
| | (4950) | (124750) | (499500) | (1999000) | (12497500) | (49995000) |
| ENS_SS | 0.0011174 | 0.0055482 | 0.020251 | 0.041009 | 0.15368 | 0.41325 |
| | (900) | (10972) | (30780) | (88080) | (350050) | (990000) |
| ENS_BS | 0.00090068 | 0.003208 | 0.011526 | 0.02105 | 0.079892 | 0.18858 |
| | (646) | (6999) | (18415) | (51171) | (195664) | (543093) |
| FMODE | 0.020339 | 0.028142 | 0.079487 | 0.14851 | 0.61336 | 1.7341 |
| | (879) | (10921) | (30704) | (87975) | (349889) | (989758) |

表 3 四种排序方法对于随机生成种群的对比测试：排序耗时秒(比较次数)

| 算法 | NP | | | | | |
|---|---|---|---|---|---|---|
| | 100 | 500 | 1000 | 2000 | 5000 | 10000 |
| NSGAII | 0.012643 | 0.30389 | 1.2565 | 5.1451 | 33.629 | 152.3857 |
| | (4950) | (124750) | (499500) | (1999000) | (12497500) | (49995000) |
| ENS_SS | 0.0007792 | 0.0068237 | 0.015762 | 0.044478 | 0.20465 | 0.56412 |
| | (1237) | (16263) | (49164) | (147723) | (627683) | (1910611) |
| ENS_BS | 0.00079018 | 0.0078485 | 0.021014 | 0.064339 | 0.32262 | 1.1166 |
| | (1451) | (22916) | (76597) | (250421) | (1179679) | (4015154) |
| FMODE | 0.012497 | 0.034995 | 0.072663 | 0.22605 | 0.96877 | 2.9161 |
| | (1395) | (16994) | (51197) | (151049) | (636641) | (1927848) |

### 3.3 总体性能对比

本节给出 FMODE 与经典算法总体性能比较，对比结果如表 4、5 所示，算法性能评价采用 IGD 指标，指标详细说明见文献[16]，表中的最优结果用粗体显示，测试结果为五次独立运行的均值。同时给出所有算法在六个测试函数上的运算时间和进化曲线，结果如图 2~7 所示。

从表 4、5 和图 2~7 的结果可以看出，四种算法在求六个函数时进化曲线大致相似，其中 ENS-BS 和 ENS-SS 两种算法稍微占优，这得益于 ENS 算法把冗余计算尽量降低。在求解 DTIZ1 和 DTLZ3 函数时，FMODE 进化用时稍多于 ENS，但是远少于 NSGAII。在求解 DTLZ4 函数表现上，FMODE 性能优于 NSGAII，所用耗时稍微差于 NSAGII。在解决 DTLZ2 和 DTLZ6 时，FMODE 表现性能与对比算法基本一致，但用时稍差与 NSGAII。特别在解决 DTLZ6 时，FMODE 用时明显较多，深入分析原因，发现在求解该问题时，大部分个体都聚集在第一等级，所以 FMODE 需要对第一等级的所有个体进行计算处理，无法体现其在计算的同时选择下一代个体的优势。算法计算得到的前端与理论前端对比见图 8 所示。

总体而言，在 IGD 评价指标性能方面，提出的 FMODE 算法的与 ENS、NSGAII 相差不大，性能表现基本相同。但是在时间对比方面，稍微差于 ENS，明显优于 NSGAII，从而验证了提出算法的有效性和正确性.

表 4 四种算法求解最小化三目标问题
DTLZ1-DTLZ6 上 IGD 均值测试对比

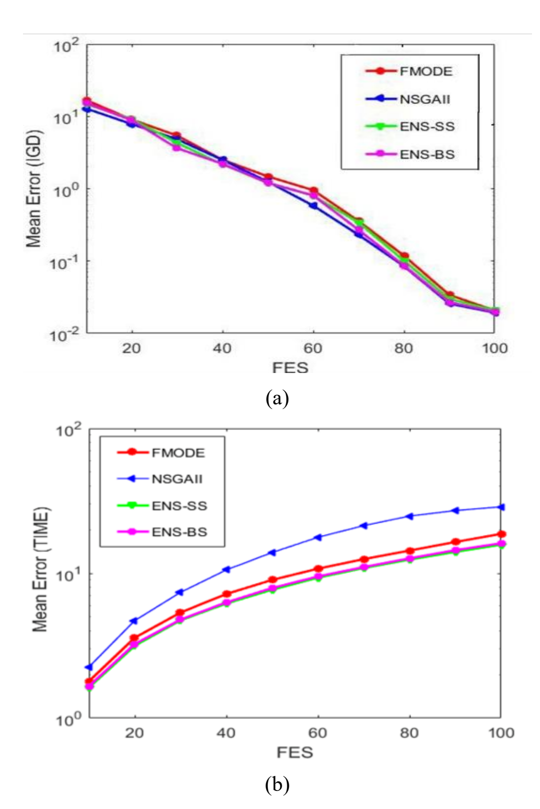
| 函数 | mean error | | | |
|---|---|---|---|---|
| | ENS_SS[11] | ENS_BS[11] | NSGAII[2] | 本章算法(FMODE) |
| DTLZ1 | 1.60E-02 | 1.62E-02 | 1.61E-02 | **1.52E-02** |
| DTLZ2 | 4.44E-02 | 4.45E-02 | 4.37E-02 | **4.30E-02** |
| DTLZ3 | 4.49E-02 | 4.46E-02 | **4.44E-02** | 4.45E-02 |
| DTLZ4 | **4.39E-02** | 4.45E-02 | 4.52E-02 | 4.54E-02 |
| DTLZ5 | 2.70E-03 | **2.57E-03** | 2.97E-03 | 2.96E-03 |
| DTLZ6 | 4.32E-02 | **4.02E-02** | 8.43E-02 | 7.71E-02 |

表 5 四种算法求解最小化三目标问题
DTLZ1-DTLZ6 进化完成所用时间均值测试对比

| 函数 | mean error | | | |
|---|---|---|---|---|
| | ENS_SS[11] | ENS_BS[11] | NSGAII[2] | 本章算法(FMODE) |
| DTLZ1 | **2.09E+01** | 2.13E+01 | 3.36E+01 | 2.56E+01 |
| DTLZ2 | 2.91E+01 | 2.71E+01 | 2.69E+01 | **2.65E+01** |
| DTLZ3 | **4.94E+01** | 5.55E+01 | 8.64E+01 | 6.11E+01 |
| DTLZ4 | 1.33E+01 | **1.30E+01** | 1.79E+01 | 1.75E+01 |
| DTLZ5 | 2.01E+01 | 2.01E+01 | 1.98E+01 | **1.80E+01** |
| DTLZ6 | 2.15E+02 | **2.06E+02** | 2.30E+02 | 2.51E+02 |

表 6 四种算法求解最小化三目标问题
DTLZ1-DTLZ6 进化完成所需代数测试对比

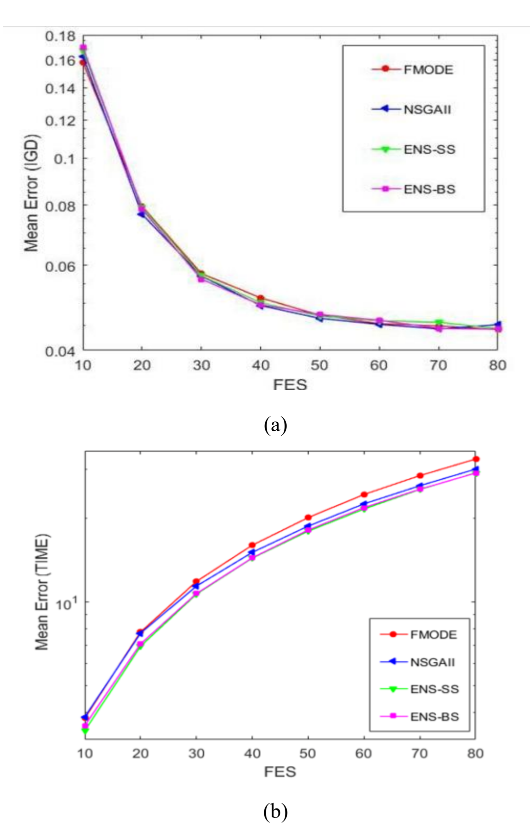
| | DTLZ1 | DTLZ2 | DTLZ3 | DTLZ4 | DTLZ5 | DTLZ6 |
|---|---|---|---|---|---|---|
| 进化代数 | 100 | 80 | 300 | 55 | 70 | 1000 |




(a)

(b)

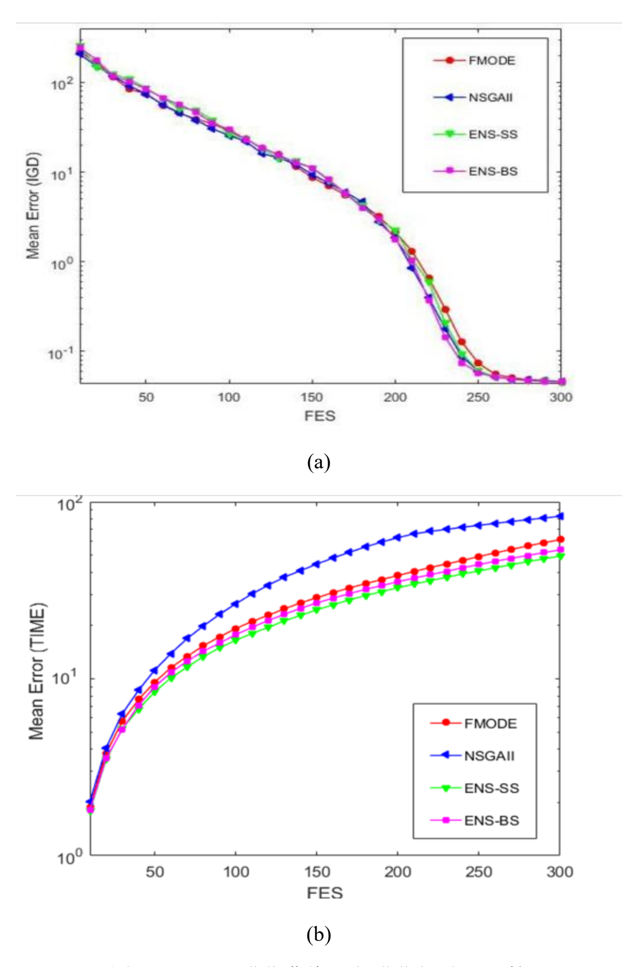
图 2　DTLZ1 进化曲线(a)与进化耗时(b)比较




(a)

(b)

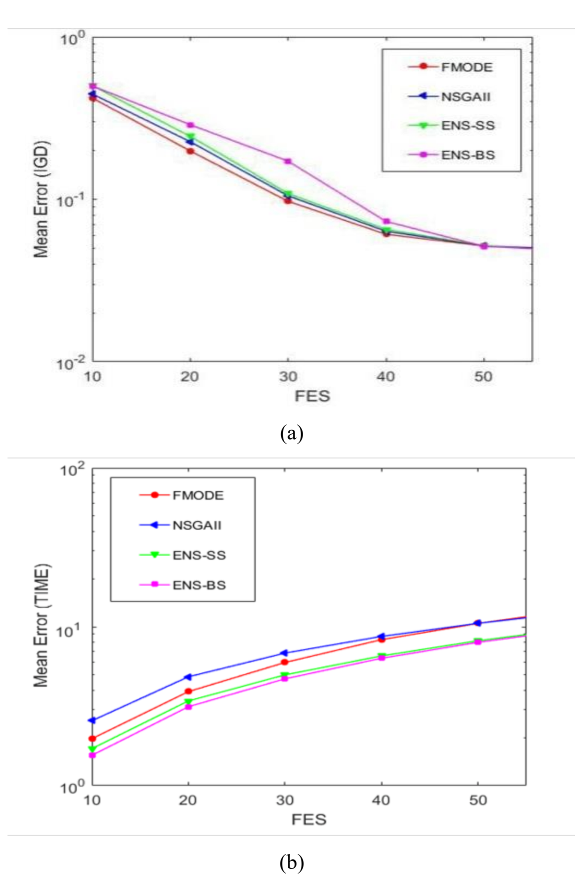
图 3　DTLZ2 进化曲线(a)与进化耗时(b)比较




(a)

(b)

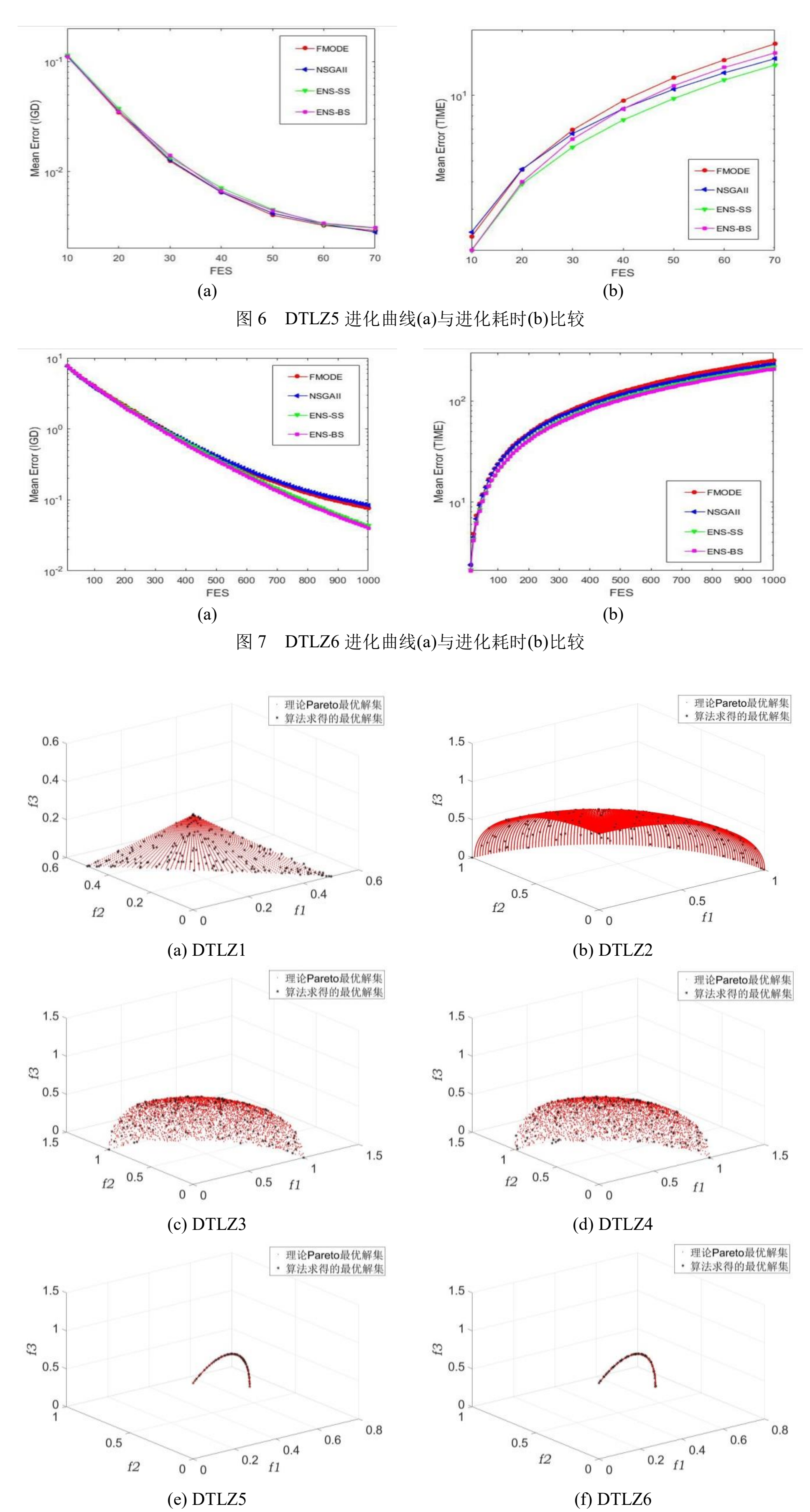
图 4　DTLZ3 进化曲线(a)与进化耗时(b)比较




(a)

(b)

图 5　DTLZ4 进化曲线(a)与进化耗时(b)比较




(a)　　　　(b)

图 6　DTLZ5 进化曲线(a)与进化耗时(b)比较

(a)　　　　(b)

图 7　DTLZ6 进化曲线(a)与进化耗时(b)比较

(a) DTLZ1　　　　(b) DTLZ2

(c) DTLZ3　　　　(d) DTLZ4

(e) DTLZ5　　　　(f) DTLZ6

图 8　DTLZ1-DTLZ6 理论前端与 FMODE 算法得到的前端对比，红色为理论前端，黑色为算法计算得到前端，(a)~(f)分别为 DTLZ1-DTLZ6

### 3.4　算法 FMODE 的参数设置

本节研究提出算法 FMODE 的参数设置，给出在不同交叉概率 Cr、变异因子 F、变异策略的组合对算法性能的影响。

在变异方式“DE/Rand/1/bin”中，变异因子 F=0.5，群体规模 NP=200，最大进化代数 FES=1000 代时，对交叉概率 Cr 进行不同的取值测试，其 IGD 均值结果如表 7 所示。通过表 7 可

以发现，“DE/Rand/1/bin”变异方式下，缩放因子 F=0.5，交叉概率 Cr=0.2 或 0.3 时效果最好，由此得到 F=0.5 和 Cr=0.2 作为经验值。在其他变异策略下，不同的交叉和缩放因子组合时，所表现的性能都差于“DE/Rand/1/bin”变异，为节省版面其数据不再给出。最终本文得出，提出算法推荐的参数设置为：“DE/Rand/1/bin”变异，缩放因子 F=0 .5，交叉概率 Cr=0.2。

表 7　DE/rand/1/bin”变异时不同参数值的 IGD 均值对比

| 缩放因子和 | mean error | | | | | |
|---|---|---|---|---|---|---|
| 交叉概率 | DTLZ1 | DTLZ2 | DTLZ3 | DTLZ4 | DTLZ5 | DTLZ6 |
| F=0.5,Cr=0.1 | 1.55E-02 | 4.15E-02 | 4.28E-02 | 4.43E-02 | 2.29E-03 | **2.15E-02** |
| F=0.5,Cr=0.2 | 1.46E-02 | **3.92E-02** | **4.01E-02** | 4.01E-02 | 2.22E-03 | 3.45E-02 |
| F=0.5,Cr=0.3 | **1.44E-02** | 3.93E-02 | 4.03E-02 | **4.00E-02** | 2.14E-03 | 6.69E-02 |
| F=0.5,Cr=0.4 | 1.52E-02 | 4.19E-02 | 4.41E-02 | 4.36E-02 | **2.03E-03** | 1.34E-01 |
| F=0.5,Cr=0.5 | 1.61E-02 | 4.52E-02 | 1.15E+01 | 4.69E-02 | 2.06E-03 | 2.60E-01 |
| F=0.5,Cr=0.6 | 1.72E-02 | 4.81E-02 | 4.86E+01 | 4.97E-02 | 2.17E-03 | 4.83E-01 |
| F=0.5,Cr=0.7 | 1.80E-02 | 4.96E-02 | 8.06E+01 | 5.25E-02 | 2.37E-03 | 8.45E-01 |
| F=0.5,Cr=0.8 | 1.26E+00 | 5.19E-02 | 1.18E+02 | 5.55E-02 | 2.48E-03 | 1.55E+00 |

## 4　结束语

本文介绍一种高效三目标非支配解排序方法，该方法能降低排序操作处理个体的数量，且在排序操作的同时,可进行选择下一代个体的操作，较大提升原始排序分配等级的计算效率。然后，将这种高效三目标非支配解排序与差分进化结合，提出基于非支配解排序的高效三目标差分进化算法，为验证算法的有效性，将提出算法与著名的 NSGAII 和 ENS 算法,在标准测试问题 DTLZl-DTLZ6 上进行实验对比，实验结果表明，提出的算法在时间复杂度和性能上都优于经典 NSGAII 算法，近似或稍差于 ENS 算法，表现出了一定的竞争力。

下一步的工作是深入分析 FMODE 存在问题，考虑结合其他进化框架或者改进参数自适应来提高算法的性能。

## 参考文献：


[1] Coello C A, Van Veldhuizen D A, Lamont G B. Evolutionary algorithms for solving multi-objective problems [M]. Berlin: Springer, 2007: 1–140.

[2] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2002, 6 (2): 182–197.

[3] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm [C]// Proc of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problem. 2002: 95-100.

[4] Gong Maoguo, Wang Shanfeng, Liu Wenfeng, et al. Evolutionary computation in China: A Literature Survey [J]. CAAI Trans on Intelligence Technology, 2016, 1 (4): 334-354.

[5] 谢涛, 陈火旺, 康立山. 多目标优化的演化算法 [J]. 计算机学报, 2003, 26 (8): 997－1003.

[6] Cui Xunxue, Li Qin, Tao Qing. Genetic algorithm for Pareto optimum-based route selection [J]. Journal of Systems Engineering and Electronics, 2007, 2 (18): 360-368.

[7] 王勇, 蔡自兴, 周育人, 等. 约束进化算法 [J]. 软件学报, 2009, 20 (1): 11-29.

[8] 郑建国, 王翔, 刘荣辉. 求解约束优化问题的 ε -DE 算法 [J]. 软件学报, 2012, 23 (9): 2374-2387.

[9] Tang Suqin, Cai Zixing, Zheng Jinhua. A fast method of constructing the non-dominated set: arena’s principle [C]// Proc of the 4th International Conference on Natural Computation. 2008: 391–395.

[10] Mc Clymont K, Keedwell E. Deductive sort and climbing sort: new methods for non-dominated sorting [J]. Evolutionary Computation, 2012, 20 (1): 1–26.

[11] Zhang Xingyi, Tian Ye, Cheng Ran, et al. An efficient approach to non-dominated sorting for evolutionary multi-objective optimization [J]. IEEE Trans on Evolutionary Computation, 2015, 19 (2): 201-213.

[12] Xu Yulong, Fang Jian’an, Zhu Wu, et al. Differential evolution using a superior-inferior crossover scheme [J]. Computational Optimization and Applications, 2015, 61 (4): 243–274.

[13] Das S, Mullick S S, Suganthan P N. Recent advances in differential evolution: an updated survey [J]. Swarm and Evolutionary Computation, 2016, 4 (27): 1-30.

[14] Abbass H, Sarker R. The Pareto differential evolution algorithm [J]. International Journal on Artificial Intelligence Tools, 2002, 11 (4): 531–552.

[15] Robic T, Filipic B. DEMO: differential evolution for multi-objective optimization [C]// Proc of the 3nd Int Conf Evolutionary Multi-criterion Computation. 2005: 520–533.

[16] 杨平, 郑金华, 李密青, 等. 一种快速构造多目标 Pareto 非支配集的方法: 选举法则 [J]. 计算机应用研究, 2009, 26 (2): 488-491.

[17] 任长安, 李智勇, 陈友文. 一种改进的基于目标空间分割的多目标进化算法 [J]. 计算机应用研究, 2010, 27 (4): 1311-1314.

[18] 吴佳英, 李平, 郑金华. 一种基于多亲遗传机制的多目标优化算法 [J]. 计算机应用与软件, 2008, 25 (2): 52-53, 79.

[19] 谭阳, 谭岳武, 唐钊轶. 基于海明差异评价的多目标进化算法 [J]. 计算机工程, 2014, (2): 212-218.

[20] 鲍培明, 朱庆保. 用于多目标进化的归一化排序非支配集构造方法 [J]. 电子学报, 2009, 37 (9): 2010-2015.

[21] 黄映, 李扬, 高赐威. 基于非支配排序差分进化算法的多目标电网规划 [J]. 电网技术, 2011, 35 (3): 85-89.

[22] 李斌, 钟润添, 肖金超, 等. 一种基于边缘分布估计的多目标优化算法 [J]. 电子与信息学报, 2007, 29 (11): 2683-2687.

[23] Wang Xianping, Tang Lixin. An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization [J]. Information sciences, 2016, 348: 124-141.

[24] Cheng Jixiang, Gary G. Yen, Zhang Gexiang. A grid-based adaptive multi-

objective differential evolution algorithm [J]. Information sciences, 2016, 367: 890-908.

[25] 李雯，李和成. 基于空间网格划分的多目标进化算法 [J]. 计算机工程与应用, 2014, (8): 53-56, 117.

[26] 许玉龙，方建安，张晗，等. 基于非支配解排序的快速多目标微分进化算法 [J]. 计算机应用, 2014, 34 (9): 2547-2551.